



Tutorial for CH5, CH6, CH7

汇编语言程序设计

第5, 6, 7章 习题

第五章 循环与分支程序设计

【思考题】

1. 编制一个汇编语言程序分哪几步？
2. 循环程序有哪两种基本结构？由几部分组成？
3. 设计算法时对可能出现的边界情况如何考虑？
4. 如何设置逻辑尺？
5. 起泡排序算法用汇编语言如何实现？
6. 如何理解数组排序算法中采用的折半查找法？
7. 如何使用跳跃表法实现CASE结构？

【学习目标】

了解并掌握循环程序的构造方法，尤其是对循环控制条件的设置以及可能出现的边界情况的考虑。

掌握起泡排序算法这种多重循环程序设计中的常用方法。交换标志位的设置在此算法中更能提高效率。

学会在数组排序算法中采用折半查找法来提高查找效率。

学会使用跳跃表法实现CASE结构。

【学习指南】

掌握编程的四个步骤至关重要。通过多看举例，学会正确分析理解题意、选择合适的数据结构及算法、坚持先画框图、选取有效指令编程、最后应当掌握运用调试手段进行调试。

学习多重循环程序设计前应熟练掌握单层循环程序设计的各种实现方法及实现细节，如对可能出现的边界情况的处理等。

学习起泡排序算法、折半查找法、跳跃表法之前，应首先理解传统实现方法。

【难重点】

循环控制条件的选择。

考虑循环算法时注意可能出现的边界情况。

静态地预置逻辑尺。动态地修改标志位。

多重循环程序设计时应分别考虑各重循环的控制条件及其程序实现，相互之间不能混淆。另外，应该注意在每次通过外层循环再次进入内层循环时，初始条件必须重新设置。

起泡排序算法是多重循环程序设计中的一种常用方法。

数组排序算法中可以采用折半查找法来提高查找效率。

CASE结构可以使用跳跃表法实现。

【知识点】

编制一个汇编语言程序的一般步骤

5.1 循环程序设计

5.1.1 循环程序的基本结构

5.1.2 循环程序设计方法举例

循环控制条件

边界情况的处理

逻辑尺

5.1.3 多重循环程序设计举例

起泡排序算法

交换标志位

5.2 分支程序设计

5.2.1 分支程序的基本结构

5.2.2 分支程序设计方法举例

折半查找法

5.2.3 跳跃表法

第五章 作业

- **5.3** 将AX寄存器中的16位数分成4组，每组4位，然后把这四组数分别放在AL、BL、CL、DL中。
- **5.7** 编制一个汇编语言程序，求出首地址为DATA的100D字数组中的最小偶数，并把它放在AX中。
- **5.12** 有一个首地址为MEM的100D字数组，试编制程序删除数组中所有为零的项，并将后续项向前压缩，最后将数组的剩余部分补上零。
- **5.13** 在STRING到STRING+99单元中存放着一个字符串，试编制一程序测试该字符串中是否存在数字.如有,则把CL的第5位置1,否则将该位置0。

- **5.14** 在首地址为TABLE的数组中按递增次序存放着100H个16位补码数,试编写一个程序把出现次数最多的数及其出现次数分别存放在AX和CX中。
- **5.15** 数据段中已定义了一个有n个字的数组M,试编写一程序求出M中绝对值最大的数,把它放在数据段的M+2n单元中,并将该数的偏移地址存放在M+2(n+1)单元中。
- **5.19** 已知数组A包含15个互不相等的整数,数组B包含20个互不相等的整数.试编制一程序,把既在A中又在B中出现的整数存放于数组C中。
- **5.24** 假设已编制好5个歌曲程序,它们的段地址和偏移地址存放在数据段的跳跃表SINGLIST中,试编制一程序,根据从键盘输入的歌曲编号1-5,转去执行五个歌曲程序中的一个。

第六章 子程序结构

【思考题】

1. 子程序调用和返回时需要保护与恢复哪些寄存器？
2. 子程序的参数传送有几种方法？
3. 伪操作STRUC的用途是什么？
4. 子程序嵌套与递归的区别？

【学习目标】

了解子程序的构成原理，学会运用多种方法在调用程序与子程序之间传递参数。掌握递归子程序的编程技巧。

【学习指南】

熟悉堆栈的运行过程对学习这一章至关重要。

【难重点】

子程序的调用和返回。保护与恢复寄存器。子程序的参数传送。伪操作STRUC的定义及应用。嵌套与递归子程序。DOS系统功能调用。

【知识点】

前言 子程序的作用和意义

6.1 子程序的设计方法

6.1.1 过程定义伪操作

6.1.2 子程序的调用和返回

6.1.3 保护与恢复寄存器

6.1.4 子程序的参数传送

通过寄存器传送参数、通过地址表传送参数地址、通过堆栈传
送参数或参数地址

伪操作STRUC的定义及应用

6.2 嵌套与递归子程序

6.2.1 子程序的嵌套

6.2.2 递归子程序

6.3 子程序举例

6.4 DOS系统功能调用

第六章 作业

- **6.4** 分析下面“6.4题的程序”的功能,写出堆栈最满时各单元的地址及内容。
- **6.6** 下面是用STRUC伪操作定义的参数表NAMELIST,
(1) 请用结构预置语句分配此结构的存储区;
(2) 编写一段程序,从键盘输入字符(用DOS功能调用)存入结构中,然后将输入的字符数送入DISPFILE单元中。

```
NAMELIST STRUC
    MAXLEN DB 100
    ACTLEN DB ?
    NAMEIN DB 100 DUP(?)
NAMELIST ENDS
```

- **6.9** 设有10个学生的成绩分别是76, 69, 81, 90, 73, 88, 99, 63, 100和80分.试编制一个子程序统计60-69,70-79, 80-89, 90-99和100分的人数并分别存放到S6,S7,S8,S9和S10单元中。
- **6.13** 给定一个正数 $N \geq 1$ 存放在NUM单元中, 试编写一段递归子程序计算FIB(N), 并将结果存入RESULT单元中。

Fibonacci书的定义如下

$$\text{FIB}(1) = 1$$

$$\text{FIB}(2) = 1$$

$$\text{FIB}(n) = \text{FIB}(n-2) + \text{FIB}(n-1) \quad n > 2$$

第七章 高级汇编语言技术

【思考题】

1. 宏汇编技术与子程序技术相比有哪些优、缺点？它们的工作方式有何差别？
2. 虚参与实参是否应一一对应？
3. 哑元表中的变元可以取何种值？
4. &操作符及%操作符如何应用？
5. 宏定义体内的程序段有标号怎么办？
6. 宏定义允许嵌套、递归调用吗？
7. IRP与IRPC之间的区别？
8. 条件汇编中有哪几种条件？
9. 高级语言结构中有哪几种形式？

【学习目标】

了解并掌握宏汇编、重复汇编及不定重复汇编、条件汇编和高级语言结构编程技术。

【学习指南】

学习宏汇编时注意与子程序技术做比较。通过多看举例，掌握哑元表变元取值、宏定义的嵌套、递归调用等难点技术。

与高级编程语言相比较地学习重复、不定重复、条件汇编以及几种高级语言结构汇编技术。

【难重点】

宏调用的优点、宏调用和子程序调用工作方式的差别

虚参与实元

哑元表中的变元取值的多样性

&操作符

%操作符

LOCAL伪操作解决宏定义体中的符号问题

宏定义的嵌套、递归

重复汇编

不定重复汇编

条件汇编中的多种条件

高级语言结构中的几种汇编形式

【知识点】

7.1 宏汇编

7.1.1 宏定义、宏调用和宏展开

宏定义体、宏指令名、哑元表、形式参数、实元表

宏调用的优点、宏调用和子程序调用工作方式的差别

7.1.2 宏定义中的参数

宏定义可以无变元

变元可以是操作码

&操作符可以把前后两个符号合并形成操作码、操作数或是一个字符串。

%操作符把跟在它之后的表达式的值转换成当前基数下的数，在宏展开期间，用这个数来取代哑元。

:= 操作符

:REQ操作符

7.1.3 LOCAL伪操作

7.1.4 在宏定义内使用宏

宏定义必须先定义后调用。允许嵌套、递归

7.1.5 列表伪操作

MASM提供 .XALL、.LALL和 .SALL控制汇编清单中宏展开的列出情况

;;伪操作符

7.1.6 宏库的建立与调用

7.1.7 PURGE伪操作

7.2 重复汇编

7.2.1 重复伪操作

7.2.2 不定重复伪操作

7.2.2.1 IRP伪操作

7.2.2.2 IRPC伪操作

7.3 条件汇编

7.4 高级语言结构

第七章 作业

➤ 7.3 ...

➤ 7.4 ...

➤ 7.7 ...

➤ 7.8 ...

➤ 7.11 试编写一段程序完成以下功能,如给定名为X的字符串长度大于5时,下列指令将汇编10次.

ADD AX,AX